

Entity Type Recognition for Heterogeneous Semantic Graphs

Jennifer Sleeman and Tim Finin and Anupam Joshi

Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
{jsleem1,finin,joshi}@cs.umbc.edu

Abstract

We describe an approach for identifying fine-grained entity types in heterogeneous data graphs that is effective for unstructured data or when the underlying ontologies or semantic schemas are unknown. Identifying fine-grained entity types, rather than a few high-level types, supports coreference resolution in heterogeneous graphs by reducing the number of possible coreference relations that must be considered. Big Data problems that involve integrating data from multiple sources can benefit from our approach when the data's ontologies are unknown, inaccessible or semantically trivial. For such cases, we use supervised machine learning to map entity attributes and relations to a known set of attributes and relations from appropriate background knowledge bases to predict instance entity types. We evaluated this approach in experiments on data from DBpedia, Freebase and Arnetminer using DBpedia as the background knowledge base.

Introduction

Big Data is often characterized as data exhibiting the four Vs: Volume, Velocity, Variety and Veracity (McAfee, Brynjolfsson, and others 2012). Annotating data elements with semantic representations can help manage two of them: variety and veracity. Often what this entails is the integration of data from different sources whose schemas are unknown, largely syntactic or very weak, are impossible or difficult to integrate. Automatically linking data to common semantic models supports integration and interoperability, especially if the semantic models support reasoning. Semantic annotations can also help ensure veracity by detecting violations of semantic constraints and allow the application of semantically grounded statistical models.

Often attempts to solve Big Data integration problems are addressed by means of schema mappings, record linkage and data fusion (Dong and Srivastava 2013). In this regard, coreference resolution becomes a necessity and often traditional approaches are not designed to solve these types of integration problems because they do not account for integrating data from multiple, and often schema-less sources.

Finding entity mentions and identifying their types are important steps in many data analysis tasks including processing structured tables and logs, semi-structured graphs and unstructured text. The results directly support subsequent tasks, such as record linkage, question answering,

coreference resolution and ontology mapping. For example, identifying medical conditions based on symptoms requires integrating medical records from a particular medical practice, known medical conditions from a trusted medical knowledge base, and possibly results from a question answering system, all of which may or may not have some underlying ontological structure. Entity resolution can support the identification of medical conditions, by identifying entities and matching entities that are likely to be coreferent. In this example medical conditions are the entities and their associated symptoms are properties. Identifying a match between a 'new instance', i.e. the patient's list of symptoms and a 'known medical condition' is an example of identifying whether they corefer to each other.

Most natural language analysis systems use linguistic evidence and context to identify entity mention strings and predict their type, typically chosen from a relatively small number of high-level possibilities, such as person, place and organization, perhaps augmented with additional application-specific types. For many forms of structured and semi-structured data (e.g., tables, logs, XML, JSON), schemas may or may not be available and, if so, are likely to be simple and semantically weak. When working with semantic graphs, if an ontology is present, the ontology may explicitly define the entity types. However, in situations where semantic graphs are not defined ontologically or when the data itself does not sufficiently use the ontology, the types are harder to identify. Linguistic analysis cannot be used in this case since it relies upon the structure of the sentence to understand the components of the sentence, which is not present when data is represented as semantic graphs or a similar representation.

When performing coreference resolution over RDF data or a similar formalism, the entity types can be explicitly given in a familiar ontology and their properties understood, enabling systems to reason about instance equality (Ferrara et al. 2008; Seddiqui and Aono 2010; Araujo et al. 2011). When this is not the case, i.e., when the ontologies are not accessible, not understood or several non-aligned ontologies are used, direct reasoning about instance equality is difficult, if not impossible. We believe that this situation will be common in many big data applications, where semantic annotations may be relatively simple and where entities and their schemas can have very different representations.

Contribution

In this work we address identifying fine-grained entity types as a pre-filter for algorithms that determine which entities in a given heterogeneous data set coref. For example, in the medical domain, for the condition cancer, we would not only identify high level cancer types such as “carcinoma”, “sarcoma”, “leukemia”, “lymphoma and myeloma”, and “central nervous system cancers” (Institute 2014). Rather we would also identify more fine-grained types such as ‘breast cancer’ and ‘bladder cancer’. Our preliminary experiments show a one-level deep identification and our ongoing work will include experiments that show identification of types at various levels.

Among linguistic-based entity recognition approaches, most of the research today does not address fine-grained entity type identification. However, it is often useful to have a more fine-grained understanding of entity types to support efforts in heterogeneous data integration. We use a linear approach for compatibility with Big Data architectures. With large knowledge bases there could exist thousands of entity types, it would be inefficient and unnecessary to evaluate an instance against each entity type. By means of information theory and high potential predicate filtering, we associate each new instance with a set of high potential candidate entity types, resulting in a significant reduction in the number of classifications. The results of our mapping approach from a single instance to a set of entity types allows us to cluster candidate coreferent instances by entity types.

Background

Semantic graphs are graph-based representations that typically are represented as triples. Resource Description Framework (RDF) is commonly used to describe resources on the Web and provides a graph-based representation (Beckett 2004; Brickley and Guha 2004).

An RDF graph is a set of triples, each of which has a subject, predicate and object. For example, for the DBpedia resource *Monaco*, *Monaco* would be the subject, an attribute such as *areaTotal* would be the predicate and a literal value *1.98* for *areaTotal* would be the object. A triple T is represented by a subject s , a predicate p , and an object o , such that $T(s, p, o)$, where o is a node, s is a node and p defines the relationship between s and o by a URI. Given a node can be a URI identifying the node, a literal, or blank, the following definitions applies: $s \in (URI \cup Blank)$, $p \in (URI)$ and $o \in (URI \cup Blank \cup Literal)$ (Beckett 2004; Brickley and Guha 2004).

Linked open data (LOD) (Bizer 2009) enables one to make data publicly available and linked to known data sets. The LOD attempts to address this problem of integrating heterogeneous data which is an inherent problem for Big Data (Bizer et al. 2012). However, linking to known data sets is a challenge, particularly when the data is heterogeneous. For example, one data set could represent an attribute using numeric values, whereas another data set could use string representations. In work by Nikolov et. al (Nikolov et al. 2009; Nikolov, Uren, and Motta 2010), they discuss the heterogeneity problem, how it relates to coreference resolu-

tion and the need for LOD automatic entity linking. Araujo et. al (Araujo et al. 2011) also references the need for an entity mapping solution that is domain independent.

Ontologies. An ontology can be thought of as a schema and provides a definition of the data, similar to an entity-relationship model (Euzenat and Shvaiko 2007). It includes a vocabulary of terms with specific meaning (Gomez-Perez, Fernandez-Lopez, and Corcho 2004). Ontologies play a critical role in the Semantic Web (Berners-Lee, Hendler, and Lassila 2001) and can be used to describe a domain. Typically ontologies use OWL (Bechhofer et al. 2004) or other languages as a representation. They define classes, instances, attributes and relations (Euzenat and Shvaiko 2007). Often one will find instance data which is described by multiple ontologies in addition to RDF. Ontologies can be privately defined and not accessible to the general public or publicly defined. It is common to be exposed to instances described by ontologies that cannot be accessed and that require an alternative method to understand the data.

Comprehensive Knowledge bases. The development of a comprehensive, general purpose knowledge base has been a goal of AI researchers dating back to the CYC project (Lenat, Prakash, and Shepherd 1985) in the early 1980s. In the past five years, two important open knowledge bases come close to realizing CYC’s vision: DBpedia and Freebase. DBpedia is a structured representation of Wikipedia (Auer et al. 2007). The DBpedia knowledge base provides classification for 3.22 million objects which mainly consists of people, locations, organizations, diseases, species and creative works (DBpedia 2013). Freebase is also a large, structured knowledge base (Bollacker et al. 2008) with considerably larger number of topics than DBpedia.

Coreference Resolution. Coreference resolution is the task of determining which instances in a collection represent the same real world entities. It tends to have an $O(n^2)$ complexity since each instance needs to be evaluated with every other instance. Various techniques have been developed to reduce the number of instances (McCallum, Nigam, and Ungar 2000; Mayfield et al. 2009; Sleeman and Finin 2010a; Rao, McNamee, and Dredze 2010; Singh et al. 2011; Uryupina et al. 2011; Song and Heflin 2011). In our previous work (Sleeman and Finin 2010b; 2010a) we also used filtering to reduce the number of candidates for the coreference resolution algorithm. We often processed data using ontologies that were not publicly available. Without an understanding of the ontologies used, it is often challenging to process data which uses those ontologies and could negatively impact accuracy.

Problem Definition

Definition 1 Given a set of instances $INST$, extracted from a set of heterogeneous sources SRC , that are not ontologically defined and not grammatically centered in a sentence, for each instance $inst_1 \dots inst_m \in INST$, we wish to associate a set of entity types $ET_1 \dots ET_m$.

Recognizing semantic graph entities is related to information extraction entity recognition, the process of recognizing entities and their type (e.g., a person, location or organization) (Ratinov and Roth 2009; Nadeau and Sekine 2007). However, it does not require the entities to be grammatically defined in a sentence structure and it entails the recognition of fine-grained entities that would be harder to obtain from a typical information extraction system.

Fundamental to our work is understanding the attributes and relations defined by the instance data. By classifying the attributes and relations, we relate unknown attributes and relations to known attributes and relations. We use this as a means for predicting entity types among heterogeneous semantic graphs.

A similar problem arises in work related to database interoperability (Nottleman and Straccia 2007; Berlin and Motro 2002; Do, Melnik, and Rahm 2003) and ontology matching (Albagli, Ben-Eliyahu-Zohary, and Shimony 2012; Mitra, Noy, and Jaiswal 2005). In both, integrating heterogeneous data drawn from different repositories with different schemas is difficult simply because it is hard to establish that an attribute or relation in one schema is the same (or nearly the same) as an attribute or relation in another (Jaiswal, Miller, and Mitra 2010).

LOD (Bizer 2009) has specifically addressed the issue of linking heterogeneous structured data in RDF to enable interoperability. In order to add an RDF dataset to a LOD collection, we represent the information as RDF and then link its elements (classes, properties and individuals) to known elements elsewhere in the collection. Though the “LOD cloud” collection has grown significantly, the total number of linked datasets is still relatively small (about 300) (Bizer, Jentzsch, and Cyganiak 2011) and the degree of interlinking often modest. Given the amount of data both available online and not available online, this number indicates that most repositories are still not linked to significant LOD collections and it is likely that these repositories use custom schemas.

Even among popular knowledge bases such as DBpedia, entity types may not always be present in the data even with a sufficiently define ontology present. Recent research by Paulheim et al. (Paulheim and Bizer 2013) found DBpedia types were only 63% complete with 2.7 million missing types.

Shvaiko et al. (Shvaiko and Euzenat 2008) described challenges in ontology matching where one such challenge is missing context. Couple the absence of context with opaquely defined attributes and often ontologies are hard to align.

Related Work

The recent work by Paulheim et al. (Paulheim and Bizer 2013) tackles the problem of identifying entity types absent from the instance data by inferring types based on existing type definitions. They assign type probabilities to indicate the likelihood of the assertion and use these as weights to establish which relations provide the best evidence for the type assertion. Their approach differs from ours in that they

use link analysis to develop their model whereas we do consider the links between graphs but we do not rely upon this alone. Rather we build a dictionary like structure that we then try to map to evaluated instances. Paulheim et al. assumes awareness of the ontologies used, we specifically address the issue of identifying types when the ontologies are either not present or insufficiently defined.

Nikolov et al. (Nikolov, Uren, and Motta 2010) describe the problem of mapping heterogeneous data where often “existing repositories use their own schemas”. They discuss how this makes coreference resolution difficult, since similarity evaluation is harder to perform when attribute mappings are unclear. They take advantage of linked data and knowledge of relationships between instances to support schema-level mappings. However, if a repository is not linked to an appropriate LOD collection, then this method is not feasible. We address this issue of custom schemas and their impact on coreference resolution by mapping attributes to a known set of attributes for various entity types.

Early work by Berlin et al. (Berlin and Motro 2002) addressed the problem of database mapping using machine learning. Their Automatch system used machine learning to build a classifier for schema-matching using domain experts to map attributes to a common dictionary. The approach performed well, achieving performance exceeding 70% measured as the harmonic mean of the soundness and the completeness of the matching process. We build on this idea, using the dictionary mapping concept which we generate from DBpedia through a process guided by information gain.

Work by (Reeve and Han 2005) provides a survey related to semantic annotation, which is more closely related to our work. They describe and benchmark methods designed for unstructured text complemented with the output of information extraction tools to construct mappings. This differs from our approach in that we start from the graphs themselves without the raw text and information extraction data and metadata. This is a key distinction since using the graphs alone is more limiting. The benchmark compared various annotation tools using annotation recall and annotation precision, which we also will use to measure our entity typing performance.

Recent research by Suchanek et al. (Suchanek, Abiteboul, and Senellart 2012) describe their approach, PARIS, for aligning ontologies. This work uses string equality and normalization measures and also takes the approach of only using positive evidence. Again our goal was to be domain-independent, such that one could use a dataset to build the dictionary of types they wish to recognize then apply our mapping process to map to these dictionaries. We use techniques more akin to traditional named entity recognition to perform the task. This distinguishes our work from much of the ontology mapping research.

Bootstrapping to a Well Defined KB

In order to assign entity types to new entity instances, we use a known knowledge base and build a model of this information. By bootstrapping to a known knowledge base, we ground unknown instance data to a known definition. For

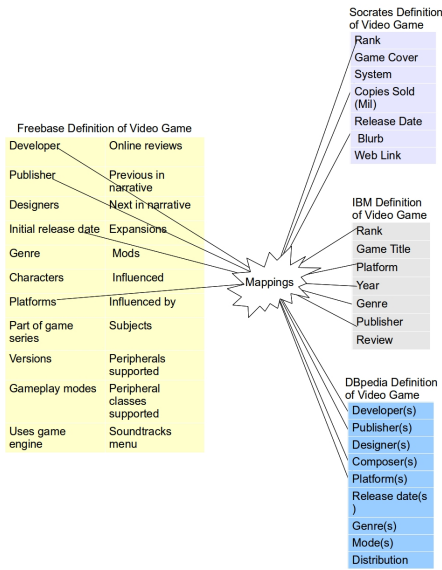


Figure 1: Mapping entity types between data sources.

example, using the medical domain, if we wanted to identify different types of leukemia, our bootstrapping knowledge would have entities, properties and relations defined that represent leukemia. If we process data that entails medical information regarding many types of cancers, we would map to our leukemia knowledge base to try to identify specific leukemia cancers. Since there are different types of leukemia, we would attempt to identify the unknown cancers with the types defined in our leukemia KB.

Definition 2 Unknown to Known Type Map - Given a set of known entity types ET extracted from a well defined KB, we create a bootstrapped system that looks to identify entity types based on ET . Each type $et_1...et_n \in ET$ is defined based on a set of predicates $EP_1...EP_n$ i.e. attributes and relations.

We bootstrap to a large well defined knowledge base to define our set of entity types. In our experiments we used DBpedia but our approach is flexible enough to work with any well defined knowledge base. We used the DBpedia ontology itself to build a model of the entity types. The model includes equivalence relationships, hierarchical relationships, and pattern similarity relationships. We use this model during evaluation to expand the number of potential entity types for a candidate and to aide in the prediction of entity types when an entity is not classified as any of the known types. Entities defined in DBpedia typically are associated with a number of entity types. This information allows us to infer types when information is not present.

Predicate Filtering

Fundamental to our work is mapping predicates associated with known types to predicates of unknown types. Figure 1 shows an example of mapping between video game definitions. However, evaluating a set of instance predicates with all possible predicates in the knowledge base is costly and

Instance Predicate	High Potential Mappings
http://creativecommons.org/ns#attributionName	http://xmlns.com/foaf/0.1/name
http://rdf.freebase.com/ns/people.person.profession	http://dbpedia.org/ontology/president http://dbpedia.org/ontology/profession
http://rdf.freebase.com/ns/award.award_winner.awards_won	http://dbpedia.org/ontology/award
http://rdf.freebase.com/ns/people.person.spouse_s	http://dbpedia.org/ontology/spouse
http://rdf.freebase.com/ns/people.person.ethnicity	http://dbpedia.org/ontology/colour
http://rdf.freebase.com/ns/music.artist.track_contributions	http://dbpedia.org/ontology/genre
http://rdf.freebase.com/ns/people.person.gender	http://dbpedia.org/ontology/gender
http://rdf.freebase.com/ns/award.award_nominee.award_nominations	http://dbpedia.org/ontology/country http://dbpedia.org/ontology/award
http://rdf.freebase.com/ns/people.person.place_of_birth	http://dbpedia.org/ontology/position http://dbpedia.org/ontology/grades http://dbpedia.org/ontology/office http://dbpedia.org/ontology/president http://dbpedia.org/ontology/ranking
http://rdf.freebase.com/ns/type.object.name	http://xmlns.com/foaf/0.1/name
http://rdf.freebase.com/ns/people.person.nationality	http://dbpedia.org/ontology/country http://dbpedia.org/ontology/colour

Figure 2: Mapping Instance Predicates.

unnecessary. Based on the model we define during bootstrapping, when we process a new instance we evaluate the instance predicates with 'high potential' predicates in our model.

Definition 3 High Potential Predicates - Given an entity type knowledge base KB , with a set of entity types ET , where each type $et_1...et_n \in ET$ is defined by $EP_1...EP_n$. From this overall distribution of predicates, we define a subset of predicates $HPEP$ and link $HPEP$ to ET .

Definition 4 Instance to High Potential Predicate Link - Given a set of instances $INST$, each instance $inst_1...inst_m \in INST$ is defined by a set of predicates $IP_1...IP_m$ which is then evaluated against $HPEP$. Each $inst_1...inst_m \in INST$ is then linked with a set of high potential candidate entity types $CET_1...CET_m$.

For example, a new instance with a set of predicates will result in the mappings in Figure 2 based on predicate filter and predicate mapping.

Entity types associated with the mapped predicates are then candidates for type matching. We are able to evaluate a smaller selection of entity types without evaluating each new instance with every other instance. Ongoing work will quantitatively show the impact of our method on computation time. We use this approach as a pre-filter to coreference resolution, reducing the number of instances that need to be evaluated without incurring a cost that is equal to the n^2 computation time cost of the coreference resolution algorithm. We can argue that this pre-filtering is beneficial to the coreference resolution algorithm because it partitions the instances into smaller clusters such that the instances within a cluster have a higher likelihood of being coreferent.

The evaluation of each instances with each potential entity type candidate results in the features used to build a supervised classification model. We perform the same mapping approach for unlabeled test data and classify these instances using the supervised model.

Feature Reduction and Information Gain

Information gain is one of the measures used to define the $HPEP$. It is also used when evaluating an instance with

a candidate entity type. In order to create the mappings defined in Figure 2, we start with a set of predicates that have high information gain. Using information gain we filter predicates that are to be mapped to instance attributes. Ongoing research will evaluate the effects of information gain thresholding as a means to filter predicates.

We calculate information gain, the reduction of entropy, for the entity type predicates. Given our set of types S and their set of predicates P , we calculate information gain and associate a weight for each predicate $\in P$.

$$Gain(S, P) = Entropy(S) - \sum_{v \in Values(P)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (1)$$

Where p is the probability of the value x_i .

$$Entropy(S) = - \sum_i^N p(x_i) \log_2 p(x_i) \quad (2)$$

Mapping

The concept of mapping to a common set of attributes is similar to database mapping and ontology alignment research (Nottelman and Straccia 2007; Berlin and Motro 2002; Mitra, Noy, and Jaiswal 2005; Albagli, Ben-Eliyahu-Zohary, and Shimony 2012). A selection of this work is discussed in more detail in Related Work.

Our work with mapping instances to types is ongoing and critical to the accuracy of the classification. Our intent is to allow for a pluggable representation whereby one can define the set of mappers that would be most appropriate for the data to be processed.

We use a distance mapper to measure the similarity of predicate labels. We currently use a Levenshtein (Levenshtein 1966) distance measure. We use a synonym mapper to measure similarity of predicate label synonyms. To obtain synonyms we use WordNet (Miller 1995) and measure similarity between sets of synonyms using Jaccard’s similarity.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

We measure similarity of predicate values by using Jaccard’s similarity and measure their frequency of occurrence and common occurrences. We also evaluate predicate values by detecting known patterns using regular expressions. For example, an email address is a commonly occurring pattern.

The results of the mappers become the features we use for classification. Ongoing work will measure statistical differences of predicate values and other properties.

Classifying Entity Types

We use a Support Vector Machine (SVM) (Joachims 2002) to develop a model for each entity type. By using a classifier per entity type, we address two important issues: we are able to use a linear classifier for this problem and we are able to horizontally scale using, for instance, a Hadoop cluster, which is relevant to big data problems. The features from

the mappers are used to create a model and that model is used to classify new instances.

Null Type and Predicting New Types

We maintain a single type which represents instances that cannot be associated with any of the known types, we call this ‘unknown’ and is akin to a ‘Null’ type. In terms of a pre-filter, our goal is to reduce the number of evaluations, however, it is reasonable to assume that a group of instances will be harder to associate with a type. We use our type model to assist us with predicting a type, when one cannot be assigned.

Each predicate is mapped to an entity type and each entity type is ontologically defined giving way to hierarchical relationships, equivalent relations, and pattern similarity relationships. We take an unfiltered approach for ‘unknown’ instances and use the ontological definitions to then find candidate types. Future work will explore this method and will aid in our work to predict new types from existing types.

Experimentation

With each experiment we randomly sampled data for training and testing and normalized the data. Our first experiment consisted of using DBpedia to define our types, DBpedia data for training and Arnetminer data (Tang, Zhang, and Yao 2007; Tang et al. 2008) for testing. Our second experiment consisted of using DBpedia to define our types, DBpedia data for training and Freebase data for testing. Our third experiment consisted of using DBpedia to define our types and DBpedia data for training and testing using two non-overlapping samples.

Performance Metrics. We use the standard precision and recall metrics for measuring performance where *TruePositive* values are those which are expected to be true and are predicted to be true, *FalsePositive* values are those predicted to be true but are actually false and *FalseNegative* are values that should be true but are predicted as false. We experimented with both stratified samples and non-stratified samples for training and testing.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

Evaluation

Our first experiment used 600 instances of the Arnetminer data set randomly selected and all of the type *Person* with each instance having an average of 11 predicates. We used the DBpedia data to build a training data set of 2000 instances with 240 entity types. When evaluating the filter performance, which is shown in Table 1, we saw 100% accuracy in its performance in designating a candidate that is consistent with the known entity type. We also were able to classify person types with close to 100% accuracy as shown in Table 2. The Arnetminer data is a sparse data set with a

Table 1: Pre-Filtering for Candidate Entity Types

Test	Found ≥ 1 Type	Found All Types
Arnetminer	100%	100%
DBpedia	99%	97%
Freebase	60%	<10%

Table 2: Arnetminer Results

Type	Precision	Recall	F-measure
person	1	.98	.99
place	1	.65	.79
organization	1	.49	.66
creativework	1	.85	.92
settlement	1	.98	.99

relatively small distribution of predicates across the data set. As a sensitivity test, we wanted to see how other classifiers classified the *Person* instances. We did find the *Place* and *Organization* classifiers had slightly lower accuracies, however, we saw higher accuracies when testing others such as *CreativeWork* and *Settlement*. There are on average only 11 attributes and there are null values for a large percentage, for example 95% of the title attribute is null.

When we experimented with the Freebase data set we used 2000 instances for training with 240 different entity types and 1000 instances for testing with over 470 different entity types. In Table 1, we show filtering results for candidate selection. What we found is that we could relax the filtering algorithm in order for us to recognize more of the potential entity types, however often it was the case that the DBpedia data set just did not contain the entity type represented in the Freebase data set. For example, 70% of the instances contain the type <http://rdf.freebase.com/ns/common.topic>. In Table 3 we show a sample of classification results. As the granularity between entity types in DBpedia is very different than Freebase, we expected to see lower than average results. We show an example of this difference in Figure 3 for entity type *Organization*.

For the DBpedia experiment we used 2000 instances for training with 240 different types, and 1000 instances for testing with 167 different entity types. There was no overlap between instances in the training and test data sets and data was sampled randomly. There were 155 overlapping entity types between the entity types in the test set and the entity types in the training sets. Since the training data and test data were

Top DBpedia Predicates for Organization	Top Freebase Predicates for Organization
http://xmlns.com/foaf/0.1/name	http://rdf.freebase.com/ns/type.object.key
http://xmlns.com/foaf/0.1/homepage	http://www.w3.org/2002/07/owl#sameAs
http://dbpedia.org/ontology/type	http://rdf.freebase.com/ns/type.object.name
http://dbpedia.org/ontology/location	http://creativecommons.org/ns#attributionURL
http://dbpedia.org/ontology/city	http://www.w3.org/1999/xhtml/vocab#license
http://dbpedia.org/ontology/genre	http://creativecommons.org/ns#attributionName
http://dbpedia.org/ontology/foundingYear	http://rdf.freebase.com/ns/common.topic.article
http://dbpedia.org/ontology/associatedBand	http://rdf.freebase.com/ns/music.record.label.artist
http://dbpedia.org/ontology/associatedMusicalArtist	http://rdf.freebase.com/ns/common.topic.webpage
http://dbpedia.org/ontology/product	http://rdf.freebase.com/ns/common.topic.official_website
http://dbpedia.org/ontology/country	http://rdf.freebase.com/ns/common.topic.image
http://dbpedia.org/ontology/keyPerson	http://rdf.freebase.com/ns/organization.organization.headquarters
http://dbpedia.org/ontology/industry	http://rdf.freebase.com/ns/organization.organization.board_members
http://dbpedia.org/ontology/hometown	http://rdf.freebase.com/ns/organization.organization.date_founded
http://www.w3.org/2003/01/geo/wgs84_pos#lat	http://rdf.freebase.com/ns/location.location.containedby

Figure 3: Top Predicates for Organization

Table 3: Sample of Freebase Entity Type Classifications

Type	Precision	Recall	F-Measure
place	0.6	0.576	0.562
person	0.635	0.629	0.625
athlete	0.336	0.376	0.337
organization	0.345	0.365	0.346
company	0.559	0.557	0.556
musical artist	0.495	0.494	0.49
architectural structure	0.478	0.477	0.473
film	0.444	0.444	0.444
building	0.612	0.61	0.609
book	0.661	0.659	0.658
soccer player	0.595	0.537	0.432
politician	0.6	0.6	0.598
event	0.361	0.371	0.356
body of water	0.446	0.444	0.444
school	0.6	0.6	0.589

Table 4: Sample of DBpedia Entity Type Classifications

Type	Precision	Recall	F-Measure
agent	0.743	0.738	0.736
person	0.781	0.774	0.773
place	0.727	0.724	0.723
populated place	0.772	0.772	0.772
settlement	0.8	0.799	0.799
work	0.843	0.838	0.838
creative work	0.843	0.838	0.838
athlete	0.805	0.798	0.797
species	0.851	0.85	0.85
eukaryote	0.746	0.74	0.738
organization	0.689	0.688	0.687
soccer player	0.895	0.893	0.893
animal	0.943	0.94	0.94
architectural structure	0.667	0.625	0.6
film	0.743	0.735	0.733
artist	0.833	0.813	0.81
album	0.778	0.733	0.722

taken from the same data set, we expected to see reasonable results. With regards to candidate filtering, as can be seen in Table 1, we often found the types expected. However, the classification results were slightly lower than our assumptions, this can be attributed to the information gain filtering and also the need to optimize the mappers. In Table 4 and in Figure 5, we show the precision, recall and f-measure scores.

What we found was when there was lower than expected performance, often the entity types in the test set were not sufficiently represented in the training set. We did not purposely try to create overlap between the training and test set. In the case of Arnetminer and the Freebase data set we are training with a completely different data set without any knowledge of type information, and therefore our preliminary results are encouraging. Our ongoing work will improve upon our existing preliminary implementation. For instance, we are currently working on statistical methods to measure the actual values of the relations and attributes since often label names cannot be mapped. We are also introducing another level of classification that is contextual.

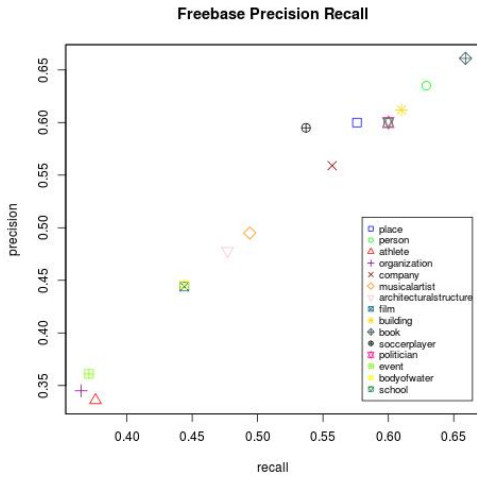


Figure 4: Freebase Precision & Recall

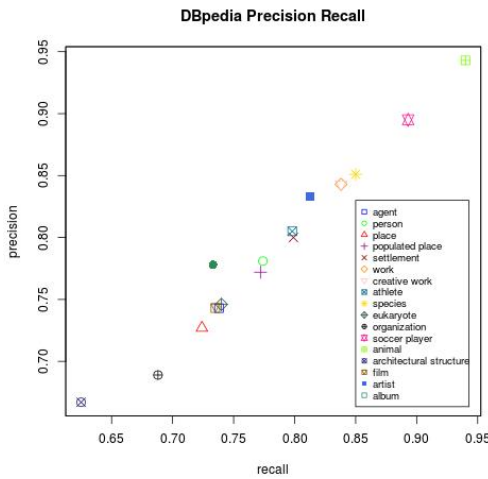


Figure 5: DBpedia Precision & Recall

Conclusions and Ongoing Work

Knowing the types of entities mentioned or referenced in heterogeneous graph data is useful and important for record linkage, coreference resolution and other tasks related to the Big Data variety and veracity problem. In Big Data problems we believe the absence of entity types is a real and continual problem.

Ideally, the data being processed is annotated with type information in an appropriate and rich semantic schema or ontology. However in many cases, such type information is absent or unavailable. This is especially common if the data has been automatically generated from a table, spreadsheet, log file or some other data format.

We have described our preliminary work for identifying fine-grained entity types. Our ongoing work will perform benchmark evaluations, will include experiments that use other data sources for bootstrapping, will include experi-

ments that show how performance is affected by relation size, and will apply our approach to a particular domain, such as the medical domain. Our ongoing work will also include adding an additional level of contextual classification, such that given a context, a certain set or entity types would become candidates for entity type recognition.

References

Albagli, S.; Ben-Eliyahu-Zohary, R.; and Shimony, S. E. 2012. Markov network based ontology matching. *Journal of Computer and System Sciences* 78.1 (2012) 105–118.

Araujo, S.; Hidders, J.; Schwabe, D.; and de Vries, A. P. 2011. Serimi- resource description similarity, rdf instance matching and interlinking. In *CoRR*, volume 1107.1104.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: a nucleus for a web of open data. In *Proc. 6th Int. Semantic Web Conf.*, 722–735. Berlin, Heidelberg: Springer-Verlag.

Bechhofer, S.; Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.; Patel-Schneider, P.; and L. Stein. 2004. Owl web ontology language reference w3c recommendation 10 february 2004. <http://www.w3.org/TR/owl-ref/>.

Beckett, D. 2004. Rdf/xml syntax specification. <http://www.w3.org/TR/REC-rdf-syntax/>.

Berlin, J., and Motro, A. 2002. Database schema matching using machine learning with feature selection. In *Proc. Conf. on Advanced Information Systems Engineering*, 452–466. Springer.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The semantic web. *Scientific American* 284(5):28–37.

Bizer, C.; Boncz, P.; Brodie, M. L.; and Erling, O. 2012. The meaningful use of big data: four perspectives—four challenges. *ACM SIGMOD Record* 40(4):56–60.

Bizer, C.; Jentzsch, A.; and Cyganiak, R. 2011. State of the LOD cloud. <http://lod-cloud.net/state/>.

Bizer, C. 2009. The emerging web of linked data. *IEEE Intelligent Systems* 24(5):87–92.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. ACM Int. Conf. on Management of Data*, 1247–1250. ACM.

Brickley, D., and Guha, R. 2004. Resource description framework (rdf) schema specification 1.0. <http://www.w3.org/TR/rdf-schema/>.

DBpedia. 2013. Dbpedia. <http://dbpedia.org/ontology/>.

Do, H.-H.; Melnik, S.; and Rahm, E. 2003. Comparison of schema matching evaluations. In *Web, Web-Services, and Database Systems*. Springer. 221–237.

Dong, X. L., and Srivastava, D. 2013. Big data integration. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE.

Euzenat, J., and Shvaiko, P. 2007. Ontology matching. Springer-Verlag, Heidelberg.

- Ferrara, A.; Lorusso, D.; Montanelli, S.; and Varese, G. 2008. Towards a benchmark for instance matching. In *Int. Workshop on Ontology Matching, volume 431, 2008*.
- Gomez-Perez, A.; Fernandez-Lopez, M.; and Corcho, O. 2004. Ontological engineering. *Springer-Verlag*.
- Han, L.; Finin, T.; and Joshi, A. 2012. Schema-free structured querying of dbpedia data. In *Conf. on Information and Knowledge Management*. ACM Press.
- Institute, N. C. 2014. National cancer institute. <https://pubs.cancer.gov>.
- Jaiswal, A.; Miller, D. J.; and Mitra, P. 2010. Uninterpreted schema matching with embedded value mapping under opaque column names and data values. *IEEE Trans. on Knowledge and Data Engineering* 291–304.
- Joachims, T. 2002. *Learning to Classify Text using Support Vector Machines Methods, Theory, and Algorithms*. Springer.
- Lenat, D. B.; Prakash, M.; and Shepherd, M. 1985. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI magazine* 6(4):65.
- Levenshtein, V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8):707–710.
- Mayfield, J.; Alexander, D.; Dorr, B.; Eisner, J.; Elsayed, T.; Finin, T.; Fink, C.; Freedman, M.; Garera, N.; Mayfield, J.; McNamee, P.; Mohammad, S.; Oard, D.; Piatko, C.; Sayeed, A.; Syed, Z.; and Weischedel, R. 2009. Cross-document coreference resolution: A key technology for learning by reading. In *Proc. AAAI Spring Symp. on Learning by Reading and Learning to Read*.
- McAfee, A.; Brynjolfsson, E.; et al. 2012. Big data: the management revolution. *Harvard business review* 90(10):60–68.
- McCallum, A.; Nigam, K.; and Ungar, L. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *The Sixth Int. Conf. on Knowledge Discovery and Data Mining*, 169–178. ACM SIGKDD.
- Miller, G. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.
- Mitra, P.; Noy, N. F.; and Jaiswal, A. R. 2005. Omen: A probabilistic ontology mapping tool. In *Int. Semantic Web Conf.*, 537–547. Springer Berlin Heidelberg.
- Mulwad, V.; Finin, T.; and Joshi, A. 2013. Semantic message passing for generating linked data from tables. In *Int. Semantic Web Con.* Springer.
- Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30.1:3–26.
- Nikolov, A.; Uren, V.; Motta, E.; and Roeck, A. 2009. Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In *Proc. 4th Asian Conf. on the Semantic Web*, volume 5926, 332–346.
- Nikolov, A.; Uren, V.; and Motta, E. 2010. Data linking: Capturing and utilising implicit schema level relations. In *Int. Workshop on Linked Data on the Web*.
- Nottleman, H., and Straccia, U. 2007. Information retrieval and machine learning for probabilistic schema matching. *Information processing and management* 552–576.
- Paulheim, H., and Bizer, C. 2013. Type inference on noisy rdf data. In *International Semantic Web Conference*.
- Rao, D.; McNamee, P.; and Dredze, M. 2010. Streaming cross document entity coreference resolution. In *Int. Conf. on Computational Linguistics*, 1050–1058.
- Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *13th Conf. on Computational Natural Language Learning*. Association for Computational Linguistics.
- Reeve, L., and Han, H. 2005. Survey of semantic annotation platforms. In *The 2005 ACM symposium on Applied computing*, 1634–1638. ACM.
- Seddiqui, M., and Aono, M. 2010. Ontology instance matching by considering semantic link cloud. In *9th WSEAS Int. Conf. on Applications of Computer Engineering*.
- Shvaiko, P., and Euzenat, J. 2008. Ten challenges for ontology matching. *On the Move to Meaningful Internet Systems: OTM 2008* 1164–1182.
- Singh, S.; Subramanya, A.; Pereira, F.; and McCallum, A. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. *Association for Computational Linguistics*.
- Sleeman, J., and Finin, T. 2010a. Computing foaf coreference relations with rules and machine learning. In *The Third Int. Workshop on Social Data on the Web*. ISWC.
- Sleeman, J., and Finin, T. 2010b. A machine learning approach to linking foaf instances. In *Spring Symposium on Linked Data Meets AI*. AAAI.
- Sleeman, J., and Finin, T. 2013. Type prediction for efficient coreference resolution in heterogeneous semantic graphs. In *Proc 7th Int. Conf. on Semantic Computing*. IEEE.
- Song, D., and Heflin, J. 2011. Automatically generating data linkages using a domain-independent candidate selection approach. In *The International Semantic Web Conference*, 649–664. Springer Berlin Heidelberg.
- Suchanek, F. M.; Abiteboul, S.; and Senellart, P. 2012. Paris: Probabilistic alignment of relations, instances, and relations. In *38th Int. Conf. on Very Large Databases*. VLDB.
- Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, 990–998.
- Tang, J.; Zhang, D.; and Yao, L. 2007. Social network extraction of academic researchers. In *ICDM'07*, 292–301.
- Uryupina, O.; Poesio, M.; Giuliano, C.; and Tymoshenko, K. 2011. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *the 24th Int. Florida Artificial Intelligence Research Society Conf.*

Autobiographical Statement

Tim Finin is a Professor of Computer Science and Electrical Engineering UMBC with over 30 years of experience in applications of AI to problems in information systems

and language understanding. His current research is focused on the Semantic Web, information extraction from text, and security and privacy. He is an AAAI Fellow, received an IEEE Technical Achievement award and is the UMBC Presidential Research Professor for 2012-15. Finin received an S.B. from MIT and a Ph.D. from the University of Illinois at Urbana-Champaign and has held full-time positions at UMBC, Unisys, the University of Pennsylvania, and the MIT AI Laboratory. He is an editor-in-chief of the Elsevier Journal of Web Semantics and a co-editor of the Viewpoints section of the Communications of the ACM.

Anupam Joshi is the Oros Family Chair Professor of Computer Science and Electrical Engineering at UMBC. He is the Director of UMBC's Center for Cybersecurity and UMBC's Cyberscholars program. He obtained a B. Tech degree in Electrical Engineering from IIT Delhi and a Masters and Ph.D. in Computer Science from Purdue University. His research interests are in the broad area of intelligent systems and networked computing with a focus on data management and security/privacy in mobile/pervasive computing environments, Semantic Web technologies and text and graph analytics, especially as applied to social media and healthcare. He has published over 200 technical papers which have lead to over 14,000 citations and an h-index of 65. He has filed and been granted several patents and his research has been supported by DARPA, NSF (including a CAREER award), DoD, NASA, IBM, Microsoft, Qualcomm, Northrop Grumman and Lockheed Martin.

Jennifer Sleeman is a Ph.D. student in Computer Science at UMBC. Her research interests include the Semantic Web, machine learning, coreference resolution and knowledge base population. She is also a Research Scientist at Leidos.